

web site, and related entertainment like music, movies or games. These additional data may be identified in a similar manner to identifying web pages based on descriptions of the data.

[0235] FIG. 14 shows an abstraction metadata list for predictive browsing according to an embodiment.

[0236] Predictive capability in a predictive browser may be provided in such a way that first content of visited web pages is pushed to the experience matrix. To do this, web page source code may be taken, cleaned from HTML and other such tags and program code, and push the remaining content to the experience matrix. This way, the random index algorithm is able to learn user's contexts.

[0237] However, using this basic method it may sometimes be difficult to get accurate predictions. There may be too much content on web pages that is not really relevant to the user context. This includes unrelated advertisements, unrelated pop-ups, and other such things common in the internet. At some internet sites, it seems even that the article that the user is interested in is well hidden behind all the unrelated content, to the irritation of the user. It is a difficult task for a software program to try to perceive the user's context from this kind of a disorder. It is demanding to programmatically separate and reveal exactly the content that is important from user's perspective from all the surrounding noise.

[0238] To overcome or alleviate these challenges, a way of creating abstraction data into an experience matrix is provided here. Abstraction data for internet sites may be by hand (or by machine), and to be fed into an experience matrix at system manufacture, startup or at periodic updates.

[0239] In this method, a person (or a machine) may select some number of most visited Internet sites, browse to those sites, and create site abstraction data (metadata) an example of which is shown in FIG. 14. The metadata may contain the URL of the site, an area or country for whose users this URL may be of importance, hyperlinks embedded in the page, company name of the website, and words that relate to the site content in an abstract level. To find the most relevant bag of words from webpage, the preprocessing may be done e.g. in a separate server which can then be queried. There may be more categories of information, like time of day for example, which will be gathered as metadata as well.

[0240] The abstraction data (metadata) may be pre-taught into an experience matrix of a user's device carrying a predictive browser. This way, the predictive browser application may use this abstraction metadata as a basis for calculating and suggesting the next website to be visited by the user. The predictive browser may use this metadata, even solely, as a basis for context calculation. This is contrary to a method where all data found on the web site is used for context calculation.

[0241] Abstraction metadata may be created for a large amount of Internet sites, as per country. The selection of which URLs to include may be based on site popularity (e.g., whether the site is among the 1000 most visited in this country.) This way it may be most probable that the user gets the benefit of the pre-taught abstraction metadata. The abstraction metadata formed in such a way may be made into a network service run on one or more of the servers of FIG. 11. Thereby, a predictive browsing system may load these abstraction metadata from the service in order to update the experience matrix. The metadata service may also provide ready-made abstraction metadata experience matrices that

can be simply added to an experience matrix in the user's device, or that can be used as such for the predictive browser.

[0242] This methodology may also contain a possibility for online updates of the metadata. In this scenario, the metadata is available at some predefined update server, from where the experience matrix implementation periodically, at user request, or asynchronously goes to fetch new metadata. This way, changes in site content can be effectively reflected to predictive browsers on client's devices.

[0243] Abstraction metadata such as in FIG. 14 may be prepared by hand or by machine. This metadata may be loaded into an experience matrix implementation in the system. This may be done for example in three ways: at assembly line during manufacturing of a user device, fetched from a predefined server when the device is for the first time connected to the internet, or at periodic or asynchronous updates during lifetime of the device. For the purpose of creating and loading data into an experience matrix, information on previous usage e.g. access of web sites may be acquired e.g. from a user account in a network service. Such service may e.g. collect such information from the user device so that it may be used in creating an experience matrix update.

[0244] Predictive browser may have configuration options for using the abstraction metadata. For example, there may be an option to ignore site content during context processing (use of experience matrix) altogether and only use abstraction metadata for the site. As another example, both the abstraction metadata and site content may be used together. As yet another example, one of the above two ways may be emphasized to some degree in context processing, that is, the experience matrix may weight e.g. abstraction metadata more than site content.

[0245] The method according to the above description may make it possible to create predictions at a predictive browser which are more meaningful. Also, the task of selecting words from web pages online during context processing may be avoided (it may be difficult to know which ones to select).

[0246] FIG. 15 shows a flow chart of predictive browsing according to an embodiment. In phase 1510, a set of words for use with an experience matrix are formed. The words may be descriptive of a context of a system, such as a current web page, calendar data, location, e-mails and so on. The experience matrix may comprise sparse vectors associated with words, and be formed such that it has data of associations and co-location of words of web pages and context information of the device in the sparse vector coefficients.

[0247] In phase 1520, at least a part of at least one sparse vector of the experience matrix is accessed to form a prediction output. This may happen as explained earlier, e.g. locally in a user device or by accessing an experience matrix in a network service

[0248] In phase 1530, suggestions of web pages may be provided to a user in response to the prediction output. For example, icons/pictures of predicted web pages may be shown to user, or titles of the web pages may be presented. The user may be able to browse the predictions e.g. by touching a touch-screen user interface. Besides touching for showing next predictions to the user, also other modalities can be used such as by saying "next" or hovering forward. The user may be able to directly browse to a web page by clicking or touching an icon of a predicted web page or in another way (such as speech) indicating the page to be accessed.

[0249] FIG. 16 shows a flow chart of predictive browsing according to an embodiment. In phase 1610, an input buffer